RL Chapter 3 - Finite Markov Decision Process

Finn Ye

Feb 2025

Introduction

1. Actions influence not only immediate rewards, but also subsequent situations.
2. Trade off between immediate and delayed reward.

Definition

1. The **agent** is the learner and decision maker.
2. The **environment** is everything the agent interacts with. The environment usually include anything that cannot be arbitrarily changed by the agent.

Setup

1. Time steps are discrete: $t = 0, 1, 2, \cdots$
2. At each step, the agent receives information on the current state $S_t \in S$ and selects their action $A_t \in A(S_t)$.
3. Depending on the action, the agent receives a reward $R_{t+1} \in R \subset \mathbb{R}$ and moves to the next state S_{t+1}.
4. The **dynamics** can be described using the probability p defined by

$$p(s', r|s, a) = P\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}$$

A sequence follows like $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \cdots$

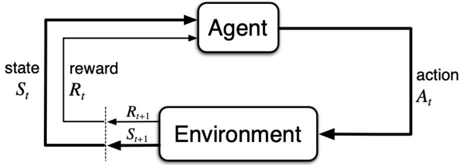

Figure 3.1: The agent–environment interaction in a Markov decision process.

Returns

1. **Episodes** are cases where there is a natural notion of final time step.
2. **Continuing Tasks** are those going on continuously without limit.
3. The agent's goal is to maximize the expected discount return:

$$G_t \equiv R_{t+1} + R_{t+2} + \cdots = \sum_{k=0}^{\infty} \delta^k R_{t+k+1}$$

   where $\delta \in [0, 1]$ is the discount rate. (I refuse to use γ to represent it.)
4. By introducing absorbing state after the terminal nodes for episodes, we can use the same notation to describe both situations.

Policies and Value Functions

1. A **policy** is a mapping from states to probabilities of selecting each possible actions. $\pi(a|s)$ describes the probability that $A_t = a$ given $S_t = s$ when the agent follows policy π.
2. The **value function** of a state s under policy π is denoted $v_\pi(s)$.
3. The value function v_π is the unique solution to its Bellman equation defined by

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \delta v_\pi(s')], \quad \forall s \in S$$

Grid World (Example 3.5)

The world is defined as a 5×5 grid. At each cell on the grid, the actions are $\{$north, south, east, west$\}$.
If the agent takes an action that will bring them off grid, their location will remain unchanged and receive a reward of -1.
Any action at state A brings the agent to A' and gives a reward of 10. Any action at state B brings the agent to B' and gives a reward of 5. All other actions give a reward of 0.

Pseudo Code for Grid World

- π : each direction is played with same probability.
- Define the world size, possible actions, and rewards.

1. Solve v_π with linear system
   - Solve for
   $$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \delta v_\pi(s')], \quad \forall s \in S$$
2. Solve v_π by value function iteration
   - Set an initial guess for v_k.
   - **Loop**
     Update v_k with Bellman equation using π given.
     If $|v_{k+1} - v_k|$ is small enough, the loop ends and we find the solution.

Pseudo Code for Grid World

- We want to find the optimal policy π^* and corresponding v_π^*.
- Set an initial guess for v_k.
- **Loop**
  Sweeping through all $a \in A$ to find the maximum value possible and update v_{k+1} with it.
  If $|v_{k+1} - v_k|$ is small enough, the loop ends and we find the optional v_π^*.
- Giving the optimal value function, sweep through possible actions to find the action that yields the optimal value. This gives the optimal policy π^*.